

Application Serial Number 10/038,547
Attorney Docket Number RA-5368

Response Filed 11/14/2005

Claim Amendments

It is requested that the following amendments be entered into the Claims:

Claims 1 - 4 (Cancelled)

1 5. (Currently Amended). A second stage lottery program for a dispatcher program
2 that dispatches tasks within an operating system of a computer system, the computer
3 system supporting at least two classes of said tasks, each of said classes including at
4 least two levels of said tasks, said dispatcher program to determine which of said tasks
5 will be assigned to a next available IP resource, said second stage lottery program
6 comprising:

7 a random number generator and selection program for generating a first random
8 number for selecting one of said at least two classes, and for generating a second
9 random number for selecting one of said at least two levels within said selected
10 class; and

11 a transfer program for ~~transferring control of~~ assigning said next available IP
12 resource from ~~said second stage lottery program to~~ execute a task assigned to said
13 selected one of said at least two levels.

1 6. (Previously Amended) The second stage lottery program of claim 5 further
2 comprising:

3 a level switching routine for handling a failure by said transfer program to find
4 a task on said selected one of said at least two levels, said level switching routine for
5 selecting a different one of said at least two levels that is associated with a task, and
6 allowing transfer of control to said task associated with said different one of said at
7 least two levels.

BEST AVAILABLE COPY

1 7. (Previously Amended) The second stage lottery program of claim 5 wherein any
2 one of said at least two levels is two times more likely to be selected than a next lower
3 one of said at least two levels.

1 8. (Currently Amended) The second stage lottery program of claim 7 wherein tasks
2 within each level of said at least two levels ~~will have tasks of like quantum values,~~
3 ~~wherein each of said quantum values defines an amount of time a task will be assigned~~
4 ~~to said next available IP resource~~ for a same amount of time as other tasks in said level.

1 9 (Previously Amended). The second stage lottery program of claim 5 wherein each of
2 said tasks has a quantum value, said quantum value identifying a computer system
3 specific amount of time in which said each of said tasks with said quantum value may
4 continuously execute on an instruction processor resource, and wherein said second
5 stage lottery employs a quantum bias routine, said quantum bias routine comprising:
6 a data capture routine for determining how much of an allotted segment of said
7 quantum value a task that has executed used before returning control to said
8 dispatcher; and
9 a bias adjustment routine for adjusting said allotted segment of said quantum value
10 based on how much of said allotted segment was used.

1 10 (Previously Amended). The second stage lottery program of claim 9 wherein said
2 bias adjustment routine does not adjust said current allotted segment to a new allotted
3 segment for said task if use of said allotted segment was interrupted by an interrupt.

1 11. (Canceled)

1 12. (Previously Amended) A computer system having a quantum timer settable to allow
2 processing on an IP resource for a limited duration of one or more tasks, also having an

BEST AVAILABLE COPY

Application Serial Number 10/038,547
Attorney Docket Number RA-5368

Response Filed 11/14/2005

3 operating system having a dispatcher program wherein all said tasks are identifiable as
4 being members of classes and wherein said dispatcher program comprises:

5 a) a scheduler code section executable to determine for how long, and to
6 which of said one or more tasks, an IP resource will be next assigned; and
7 b) a scheduler queue from which said one or more tasks may be
8 addressable and assignable to said IP resource, wherein

9 said scheduler code section has a two stage lottery execution algorithm, a
10 first of said two stages using a lottery process for determining from which class of
11 said classes a next one of said one or more tasks will be selected, said second of
12 said two stages using a second lottery process to select a task that will next be
13 assigned to said IP resource from a class determined by said first of said two
14 stages.

1 13. (Previously Amended) The computer system set forth in claim 12 wherein said first
2 stage chooses a class randomly from among all said classes using a bias settable by a
3 user.

1 14. (Previously Amended) The computer system set forth in claim 12 wherein the
2 number of said classes is selectable by a user.

1 15. (Previously Amended) The computer system as set forth in claim 12, wherein if any
2 one of said any tasks is of a class "above the lottery line" then said any one of said any
3 tasks is assigned an IP resource prior to running said scheduler code section.

1 16. (Original) The computer system as set forth in claim 12, wherein if the scheduler
2 code section's first of two stages selects a class which is empty of said any tasks, said
3 scheduler code section next chooses another class of available classes.

Application Serial Number 10/038,547
Attorney Docket Number RA-5368

Response Filed 11/14/2005

1 17. (Original) The computer system as set forth in claim 12, wherein if none of the
2 classes has any of said any tasks, said scheduler code selects a very low priority level
3 operating system task.

1 18. (Currently Amended) The computer system as set forth in claim 12, wherein said
2 second stage lottery comprises:

3 a random number generator and selection program for generating a random number
4 and for selecting a one of said at least two levels within said selected class based upon
5 a correspondence of said thereby generated random number; and

6 a transfer program for causing said IP resource to begin execution of transferring
7 ~~control from said second stage lottery program to a task found on~~ assigned to said
8 selected one of said at least two levels.

1 19. (Previously Amended) A method for use by a dispatcher algorithm in an
2 operating system in a computer system for selecting a task to provide with an available
3 instruction processor resource, wherein the task may be selected from one or more
4 classes, each of which may have one or more priority levels, wherein said method
5 comprises:

6 determining whether within a scheduler queue there are any task pointers
7 within said priority levels of said classes of tasks on said scheduler queue, and if
8 so, determining whether any said any task pointers are above a second stage
9 lottery line, and if so assigning a first of said any tasks indicated by said any task
10 pointers above said second stage lottery line to said available instruction
11 processor resource; but if not,

12 and if there is only one said priority level having any of said any task
13 pointers, assigning a first of said any tasks corresponding to said any task
14 pointers at said only one said priority level to said available instruction processor

Application Serial Number 10/038,547
Attorney Docket Number RA-5368

Response Filed 11/14/2005

15 resource; else,
16 running a first stage of a second stage lottery algorithm to select one of
17 said classes, and running a second stage of said second stage lottery algorithm
18 to select from said one of said classes a priority level from which to select said
19 task to provide to said available instruction processor resource.

BEST AVAILABLE COPY

1 22. (Previously Amended) The method of claim 21, and further comprising:
2 determining whether said each task priority level should be changed based on
3 said last time said each task was assigned to said available instruction processor
4 resource whereby

5 if said last time did not complete within said task assigned quantum, lower said
6 task priority level for said task by one priority level and raise said value of said task
7 assigned quantum by a factor of 2,

8 if said last time used less than said value of said task assigned quantum but
9 more than a small portion of said task assigned quantum, leave said priority level and
10 said small portion the same, and

11 if said last time used less than said small portion of said value of said task
12 assigned quantum, increase said priority level of said task by one and halve said task
13 assigned quantum value of said task.

1 23. (Previously Amended) The method of claim 21 wherein said task assigned
2 quantum is assigned for said each task prior to a said task pointer being on said
3 scheduler queue.

1 24. (Original) The method of claim 21 wherein a first stage lottery algorithm selects
2 which of said classes of tasks on said scheduler queue will have tasks assignable to
3 said available instruction processor resource.

1 25. (Original) The method of claim 21 wherein a first stage lottery algorithm selects
2 which of said classes of tasks on said scheduler queue will have tasks assignable to
3 said available instruction processor resource, but only if there are no tasks in classes
4 above a first lottery line, and if there are tasks in said classes above said first lottery

BEST AVAILABLE COPY

Application Serial Number 10/038,547
Attorney Docket Number RA-5368

Response Filed 11/14/2005

5 line, providing said available instruction processor resource to said tasks in classes
6 above said first lottery line in a priority order.

1 26. (Previously Amended) A dispatcher system for use within a computer system,
2 said computer system having an available instruction processor resource and a
3 scheduler for storing pointers to tasks to be assigned quantum of said instruction
4 processor resource, said dispatcher system comprising:
5 a first stage of a two stage lottery system for using a random number generator
6 to select one of multiple classes of tasks; and
7 a second stage of said two stage lottery system for selecting a task from said
8 selected one of said multiple classes, whereby said selected task is assigned to an
9 available instruction processor resource for a portion of a quantum assigned to said
10 selected task.

BEST AVAILABLE COPY